This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1.   (Currently Amended) A debugger for debugging any of a plurality of debuggees, each debuggee having a debugging type attribute selected from a plurality of debugging type attributes and representative of a type of debugging to be performed with respect to the debuggee, each debuggee also having a processor attribute selected from a plurality of processor attributes and representative of a type of processor associated with the debuggee, the debugger being instantiated on a computer and comprising:

a single debugger engine for performing debugging functions with respect to any of the plurality of debuggees, the engine including:

a plurality of debugging type blocks, each debugging type block for supporting at least one of the plurality of debugging type attributes; and

a plurality of processor blocks, each processor block for supporting at least one of the plurality of processor attributes,

wherein a particular debugging type block and a particular processor block are selected for debugging a particular debuggee based on the debugging type attribute and processor attribute of the particular debuggee,

wherein the plurality of debugging type blocks are organized into a debugging type abstraction available to provide debugging type services that vary in implementation for each debugging type,

wherein the debugging type abstraction comprises programming code, and wherein at least a portion of the programming code for the debugging type abstraction is

common as between at least some debugging type blocks and is shared by such debugging

type blocks, [[and]]

wherein the programming code for the debugging type abstraction is organized

into a tree form with generic code at a base node and more specific levels of code branching

out at nodes therefrom, each debugging type block including at least one node from the tree,

wherein the plurality of processor blocks are organized into a processor

abstraction available to provide processor services that vary in implementation for each

processor,

wherein the processor abstraction comprises programming code, and wherein

at least a portion of the programming code for the processor abstraction is common as

between at least some processor blocks and is shared by such processor blocks, and

wherein the programming code for the processor abstraction is organized into

a tree form with generic code at a base node and more specific levels of code branching out at

nodes therefrom, each processor block including at least one node from the tree.

2   (Canceled)

3.   (Previously Presented)   The debugger of claim 1 wherein the

debugging services include services selected from a group consisting of accessing memory,

accessing context, accessing system information, inserting a breakpoint, removing a

breakpoint, controlling execution, and combinations thereof.

4-6   (Canceled)

7.   (Currently Amended) The debugger of claim [[6]] 1 wherein the

processor services include services selected from a group consisting of recognizing particular

processor instructions, recognizing processor states, maintaining hardware breakpoints,

assembling code for the processor, disassembling code from the processor, disassembling

code from a dump file produced by the processor, and combinations thereof.

8-9   (Canceled)

10.   (Original)   The debugger of claim 1 wherein the engine further

includes a high level portion for issuing generic requests to the selected debugging type block

and to the selected processor block to accomplish debugging actions.

11.   (Original)   The debugger of claim 10 wherein the plurality of

debugging type blocks are organized into a debugging type abstraction available to provide

debugging type services that vary in implementation for each debugging type, wherein the

plurality of processor blocks are organized into a processor abstraction available to provide

processor services that vary in implementation for each processor, and wherein the high level

portion issues generic requests to the debugging type abstraction and to the processor

abstraction to accomplish debugging actions.

12-13 (Canceled)

14.   (Original)   The debugger of claim 1 wherein the plurality of processor attributes supported by the processor blocks include processor attributes representative of members selected from a group consisting of an X86 processor family, an ALPHA processor family, an IA64 processor family, and combinations thereof.

15.   (Original)   The debugger of claim 1 wherein the debugger further has an executable for being executed by a user, for calling the engine, and for providing an interface between the user and the engine.

16.   (Original)   The debugger of claim 15 wherein the executable includes an attribute that results in the selection of a particular debugging type block in the engine.

17.   (Original)   The debugger of claim 15 wherein the executable includes an attribute that results in the selection of a particular processor block in the engine.

18.   (Original)   The debugger of claim 1 wherein the particular debuggee is a dump file produced by a processor operating a particular mode, wherein the debugging type attribute of the dump file corresponds to the particular mode, and wherein the particular debugging type block of the engine selected for debugging the dump file supports the debugging type attribute of the dump file.

19.   (Original)   The debugger of claim 1 wherein the particular

debuggee is a dump file produced by a type of processor, wherein the processor attribute of

the dump file corresponds to the type of processor, and wherein the particular processor block

of the engine selected for debugging the dump file supports the processor attribute of the

dump file.

20.   (Currently Amended) A computer having a debugger instantiated

thereon for debugging any of a plurality of debuggees, each debuggee having a debugging

type attribute selected from a plurality of debugging type attributes and representative of a

type of debugging to be performed with respect to the debuggee, each debuggee also having a

processor attribute selected from a plurality of processor attributes and representative of a

type of processor associated with the debuggee, the debugger comprising:

a single debugger engine for performing debugging functions with respect to

any of the plurality of debuggees, the engine including:

a plurality of debugging type blocks, each debugging type block for

supporting at least one of the plurality of debugging type attributes; and

a plurality of processor blocks, each processor block for supporting at least

one of the plurality of processor attributes,

wherein a particular debugging type block and a particular processor block are

selected for debugging a particular debuggee based on the debugging type attribute and

processor attribute of the particular debuggee,

wherein the plurality of debugging type blocks are organized into a debugging

type abstraction available to provide debugging type services that vary in implementation for

each debugging type,

wherein the debugging type abstraction comprises programming code, and

wherein at least a portion of the programming code for the debugging type abstraction is

common as between at least some debugging type blocks and is shared by such debugging

type blocks, [[and]]

wherein the programming code for the debugging type abstraction is organized

into a tree form with generic code at a base node and more specific levels of code branching

out at nodes therefrom, each debugging type block including at least one node from the tree,

wherein the plurality of processor blocks are organized into a processor

abstraction available to provide processor services that vary in implementation for each

processor,

wherein the processor abstraction comprises programming code, and wherein

at least a portion of the programming code for the processor abstraction is common as

between at least some processor blocks and is shared by such processor blocks, and

wherein the programming code for the processor abstraction is organized into

a tree form with generic code at a base node and more specific levels of code branching out at

nodes therefrom, each processor block including at least one node from the tree.

21-26 (Canceled)

27.   (Original)   The computer of claim 20 wherein the engine further includes a high level portion for issuing generic requests to the selected debugging type block and to the selected processor block to accomplish debugging actions.

28.   (Original)   The computer of claim 27 wherein the plurality of debugging type blocks are organized into a debugging type abstraction available to provide debugging type services that vary in implementation for each debugging type, wherein the plurality of processor blocks are organized into a processor abstraction available to provide processor services that vary in implementation for each processor, and wherein the high level portion issues generic requests to the debugging type abstraction and to the processor abstraction to accomplish debugging actions.

29.   (Original)   The computer of claim 20 wherein the debugger further has an executable for being executed by a user, for calling the engine, and for providing an interface between the user and the engine.

30.   (Original)   The computer of claim 29 wherein the executable includes an attribute that results in the selection of a particular debugging type block in the engine.

31.   (Original)   The computer of claim 29 wherein the executable includes an attribute that results in the selection of a particular processor block in the engine.

32-50   (Canceled)